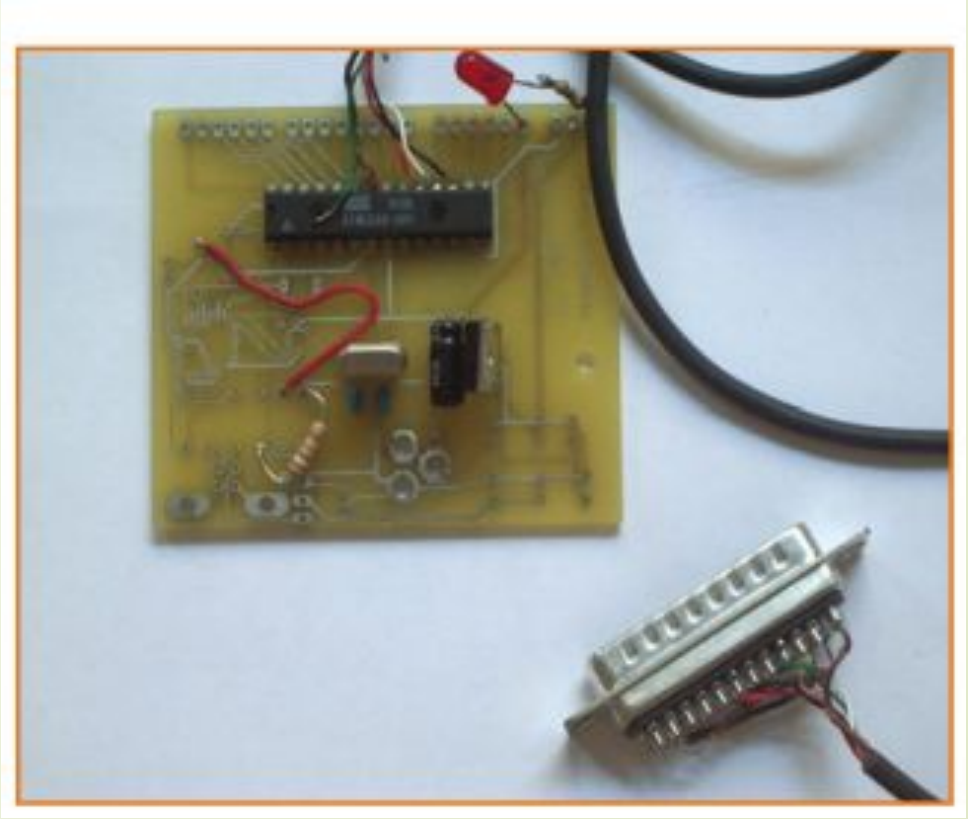
Все что вы хотели знать об Ардуино

Как сориентироваться в океане «Ардуино»

История о том, как пятеро друзей создали маленькую плату, которая взяла штурмом мир электронных самоделок

Программировнаие микроконтроллера было на языке Basic



Ядро команды Arduino (слева направо): Дэвид Куартилльз (David Cuartielles), Джанлука Мартино(Gianluca Martino), Том Иго (Tom Igoe), Дэвид Мелис (David Mellis), и Массимо Банци (Massimo Banzi) на конференции Maker Faire в Нью-Йорке (Фото: Рэнди Зильберман Клетт)

Первая плата выпущена в 2005 году

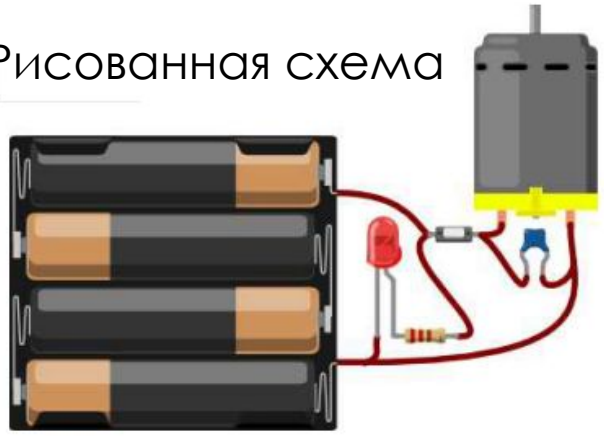
Сегодня Ардуино стал «мозгами создателей роботов»

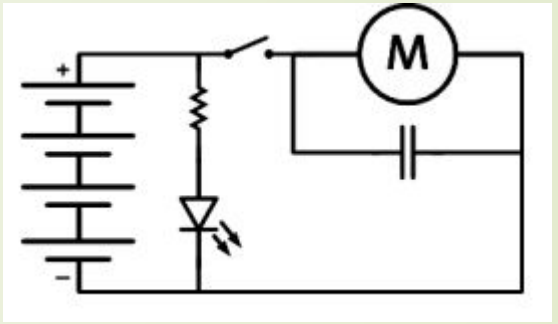
Вы можете рассматривать аппаратное обеспечение как часть культуры, которой хотите поделиться с другими людьми

- Один из первых проектор Ардуино: самодельный будильник , свисающий с потолка

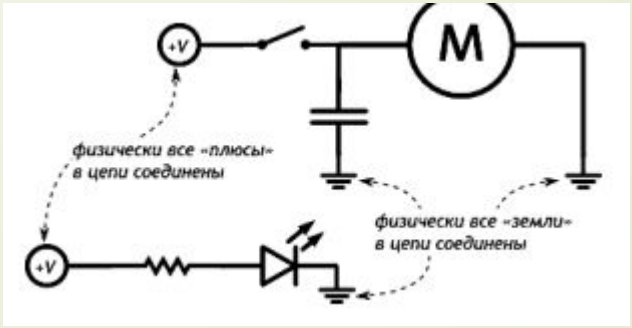
Для сборки проектов на Ардуино используют электрические схемы

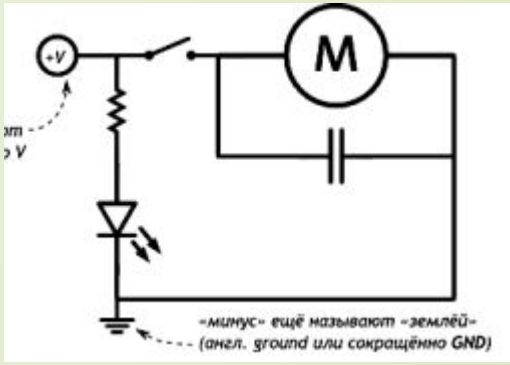
Рисованная схема





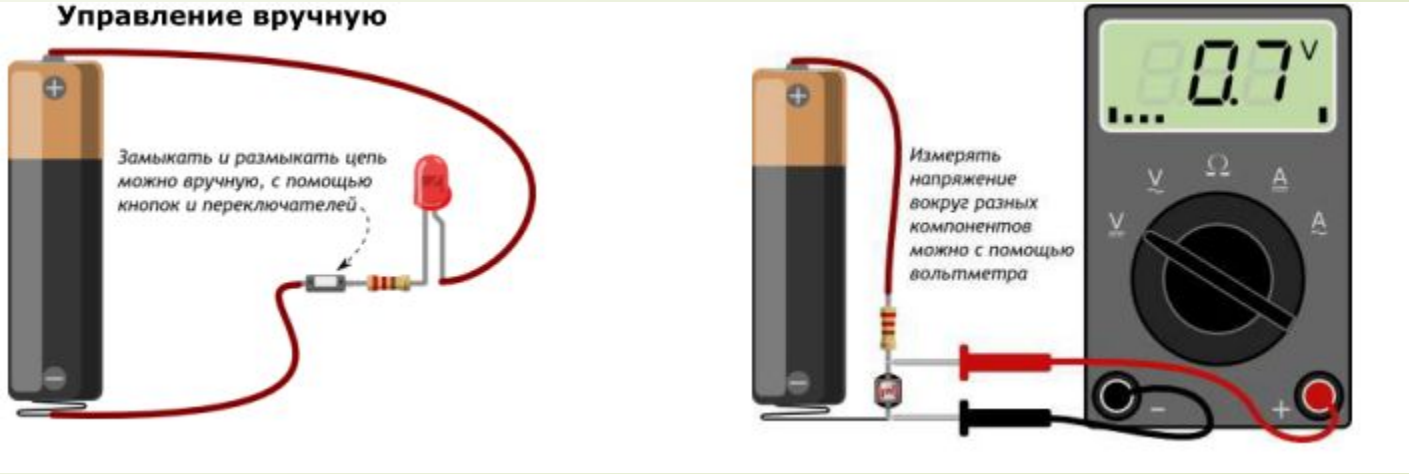
Принципиальные схемы





Управление электричеством

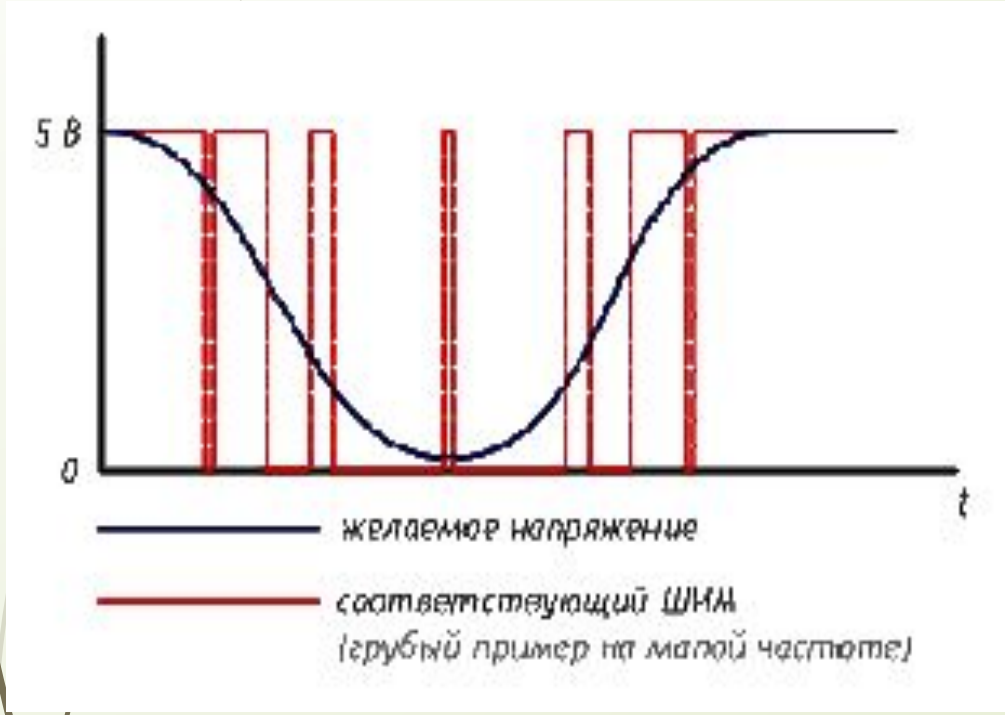
- **Если постоянно и монотонно трансформировать электроэнергию в другую форму, область применения электричества будет сильно ограничена. Огромный мир разнообразных полезных устройств открывается, если научиться контролировать и взаимодействовать с электричеством. Для этого существует несколько способов.**



Автоматическое управление

- Замыкать и размыкать цепь, измерять напряжение также можно, не вручную, а автоматически, по заданному алгоритму при помощи запрограммированного микроконтроллера.
- Типичным представителем этого семейства являются платы Arduino.

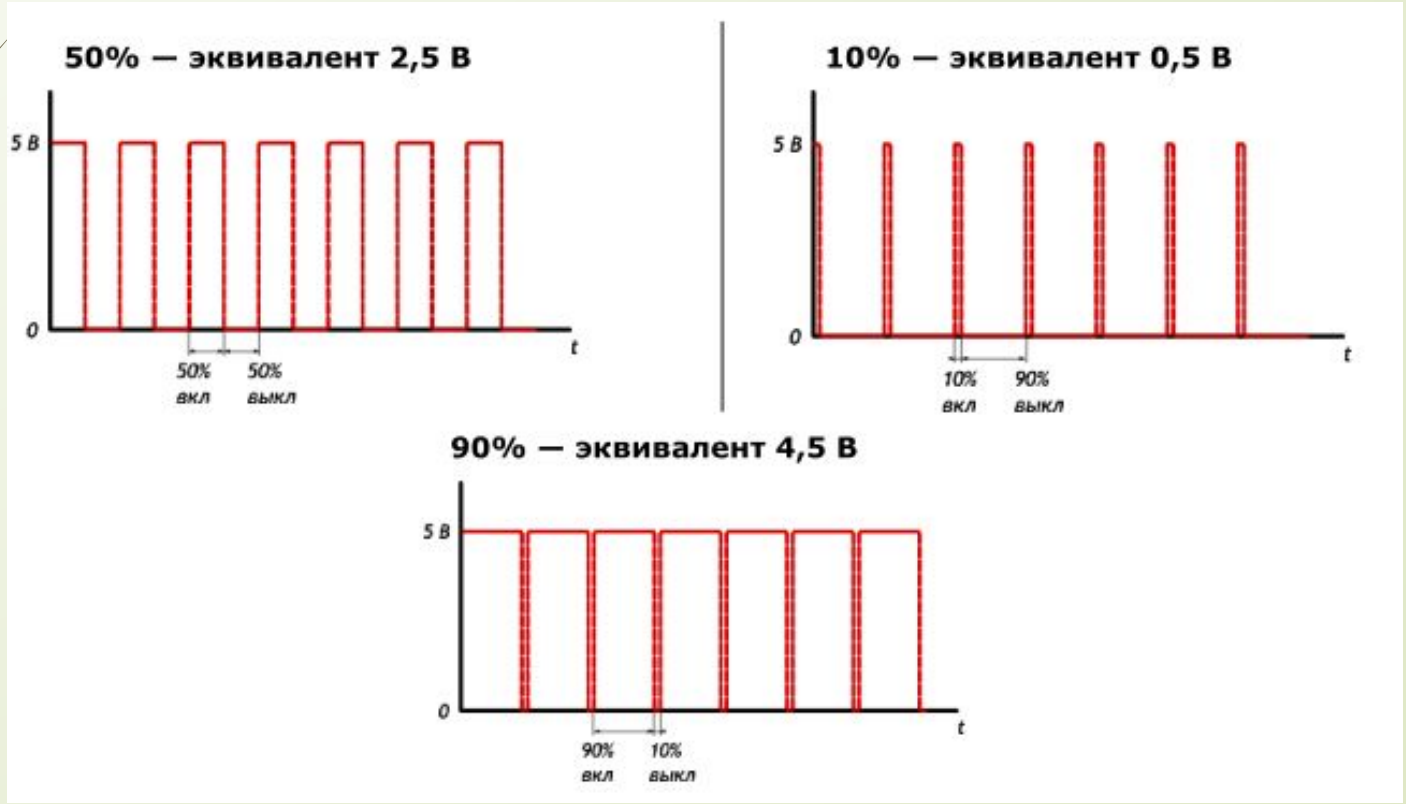
Широтно-импульсная модуляция



- Микроконтроллеры обычно не могут выдавать произвольное напряжение. Они могут выдать либо напряжение питания (например, 5 В), либо землю (т.е. 0 В) Но уровнем напряжения управляется многое: например, яркость светодиода или скорость вращения мотора. Для симуляции неполного напряжения используется ШИМ (Широтно-Импульсная Модуляция, англ. Pulse Width Modulation или просто PWM)

Скважность

- Отношение времени включения и выключения называют **скважностью** (англ. duty cycle). Рассмотрим несколько сценариев при напряжении питания Vcc равным 5 вольтам.



Пины для ШИМ

- **Не любой порт Arduino поддерживает широтно-импульсную модуляцию, если вы хотите регулировать напряжение, вам подойдут пины, помеченные символом тильда «~». Для Arduino Uno это пины 3, 5, 6, 9, 10, 11**

Особенности -1 программирования на Ардуино

- Идентификаторы переменных, констант, функций (в этом примере идентификатор LED_PIN) являются одним словом (т.е. нельзя создать идентификатор LED PIN).
- Идентификаторы могут состоять из латинских букв, цифр и символов подчеркивания _ . При этом идентификатор не может начинаться с цифры.
- Регистр букв в идентификаторе ИМЕЕТ значение. Т.е. LED_PIN , LED_pin и led_pin с точки зрения компилятора — различные идентификаторы
- Идентификаторы, создаваемые пользователем, не должны совпадать с предопределенными идентификаторами и стандартными конструкциями языка;
- если среда разработки подсветила введенный идентификтор каким-либо цветом, замените его на другой

Особенности-2 программирования на Ардуино

- Директива #define просто говорит компилятору заменить все вхождения заданного идентификатора на значение, заданное после пробела (здесь 9), эти директивы помещают в начало кода. В конце данной директивы точка с запятой ; не допустима
- Названия идентификаторов всегда нужно делать осмысленными, чтобы при возвращении к ранее написанному коду вам было ясно, зачем нужен каждый из них
- Также полезно снабжать код программы комментариями: в примерах мы видим однострочные комментарии, которые начинаются с двух прямых слэшей // и многострочные, заключённые между /*

Особенности-3 программирования на Ардуино

- Функция analogWrite(pin, value) не возвращает никакого значения и принимает два параметра:
- o pin — номер порта, на который мы отправляем сигнал
- o value — значение скважности ШИМ, которое мы отправляем на порт. Он может принимать целочисленное значение от 0 до 255, где 0 — это 0%, а 255 — это 100%

«Помигать светодиодом».

```
void setup()  // обязательная процедура
{
  pinMode (13, OUTPUT); // светодиод на пин 13
// направление пин 13 на выход - OUTPUT
}
 void loop()  // задается тело цикла
{
  digitalWrite (13, HIGH); // на пин 13 подать «1»
  delay (1000); // ждать  1 с
  digitalWrite (13, LOW); // на пин 13  подать «0»
  delay (1000);
}
```

Объявление переменных

int ledPin = 13;

// сообщает что на пин 13 будет подключена переменная ledPin

// в случае смены пина достаточно только в одном месте программы поменять значение

Мигаем светодиодом объявленной переменной

```
int ledPin = 13;
 void setup()
{
  pinMode (ledPin, OUTPUT);
}
 void loop()
{
  digitalWrite (ledPin, HIGH);
  delay (1000);
  digitalWrite (ledPin, LOW);
  delay (1000);
}
```

Функции в программе

- **digitalWrite (ledPin, HIGH)** устанавливает заданный вывод в состояние с высоким уровнем, то есть включает вывод.
- **digitalWrite (ledPin, LOW)** устанавливает заданный вывод в состояние с высоким уровнем, то есть включает вывод.
- **delay (1000**) означает паузу в 1000 миллисекунд или 1 секунду.